**第八章 Flash的测试**

**摘要：**本章的目的不是提供关于 Flash 闪存的完整测试理论，而是展示和分析Flash测试的关键方面以及提高其可测性的工具和方法，同时给出测试流程的概念，以及它与Flash 所达到的卓越质量和可靠性的关系。与测试成本和生产力相关的方面同样会被提及。

本文从Flash制造商的角度出发，以实用的术语进行处理，旨在让非专业的读者也能了解以上方面。

本章指的是NOR架构，由Fowler-Nordheim隧穿擦除，由沟道热载流子效应编程的主流Flash技术，不过其大多数方面可能也适用于其他Flash技术。

测试已知良好芯片（Known-Good-Die）、闪存卡或嵌入式闪存等主题没有被提及，否则每个主题都需要一个专门的章节来描述。

# 8.1 简介

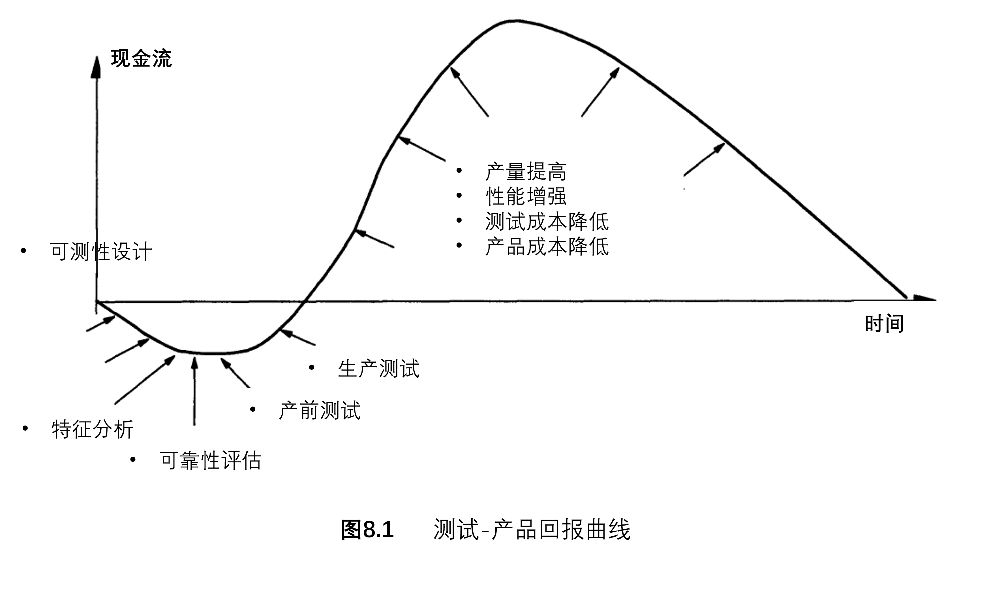
## 8.1.1 测试对生产成本的影响

在讨论测试的技术层面之前，它的经济层面同样值得一提。半导体工业总是需要巨量的投资：在生产设施和设备上进行高投入才能带来高回报。对于Flash的生产，在测试上的投资约占总投资的10%，远大于封装所需的投资。测试约占成品成本的5-15%，受生产规模、产品和制程成熟度、复杂度等的影响较大。

产品交付时间也受测试影响：晶圆阶段和封装后测试，在Flash成品生产周期中可能占1到2周时间。

## 8.1.2 测试对产品生命周期的影响

测试十分有必要，各项测试贯穿了Flash产品的整个生命周期（图8.1）。在产品开发阶段就必须考虑可测性的设计，这样产品的设计和测试开发就能同时进行。特定的测试硬件和软件能帮助设计进行调试和验证，而这通常需要设计和测试工程师的合作。



产品合格前需要通过大量的特性分析、可靠性评估和小规模生产验证。在产品开发和验证阶段进行的测试工作（广义）的质量对产品的市场投放时间、量产时间和经济收益都有很大影响。

在量产阶段，高竞争压力的Flash市场依然要求产品不断改进测试来提升良率、改善性能、消除缺陷、降低测试的总体成本。与其他所有存储器一样，Flash产品可能会被微缩或者移植到更先进的技术中，这就要求产品在开发-验证-产品化的循环中不断重复更新。

## 8.1.3 生产测试的目标

生产测试实际是一种顺序筛查，它保证出厂的产品满足性能、质量、可靠性等方面的诸多要求。它还有一些附带目标，比如收集数据以更好地控制和改进整个生产流程并提升良率。生产测试还能对同一批产品按性能进行分类（速度、Vcc范围、温度范围、擦除和编程性能等）。与其他存储器相似，绝大多数Flash产品通过冗余替换来提高良率，因此生产测试也有缺陷检测、诊断与故障修复的目标，以此更好地利用冗余资源。

以上所有方面都影响着整体测试成本和所需的投资，并受产品周期时间的严格限制。

## 8.1.4 测试对质量和可靠性的影响

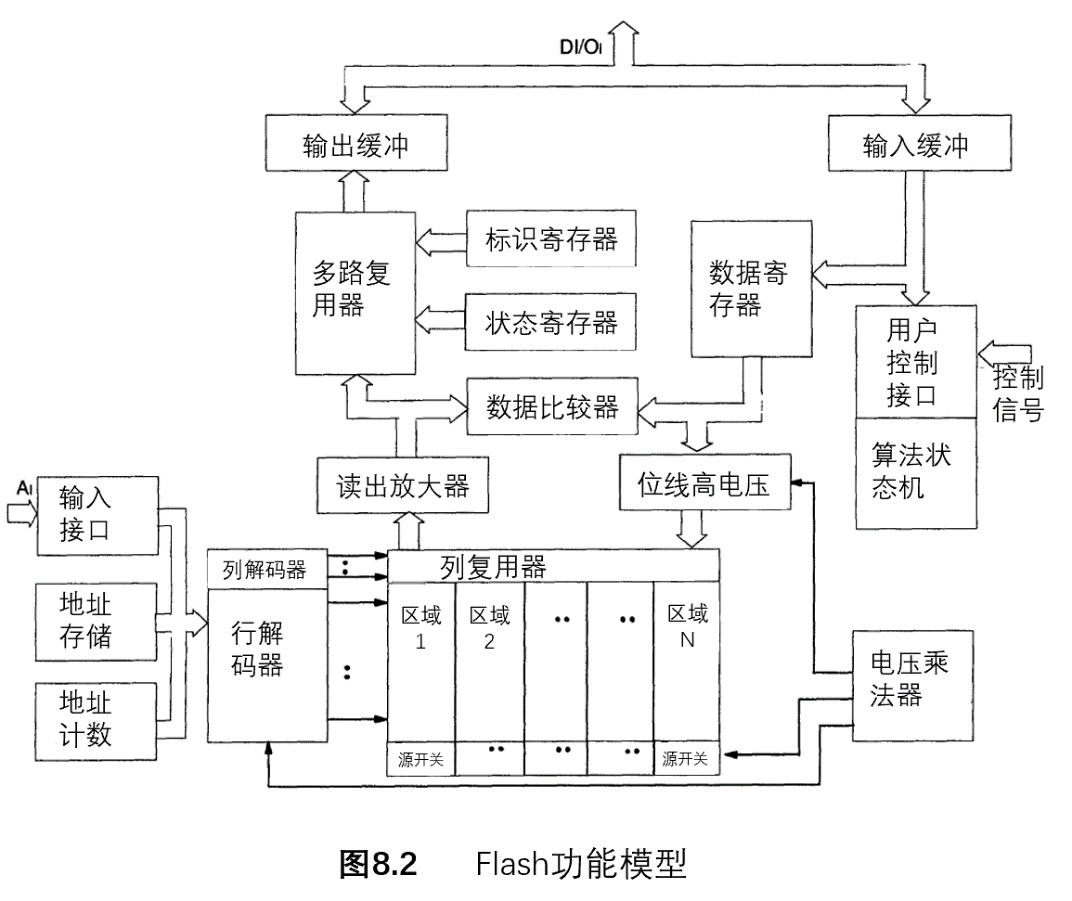
生产测试不是保证Flash产品质量和可靠性的决定性因素：它只是其中一环。其他影响因素包括：

* Flash 单元 (cell)、工艺和产品的设计；
* 工艺和产品的调试和特性分析；
* 生产资料：设备、资源、原料等；
* 整体工艺控制；
* 工艺和产品的可靠性分析；
* 制造商的技术掌握。

# 8.2 Flash测试方面

## 8.2.1 Flash功能模型

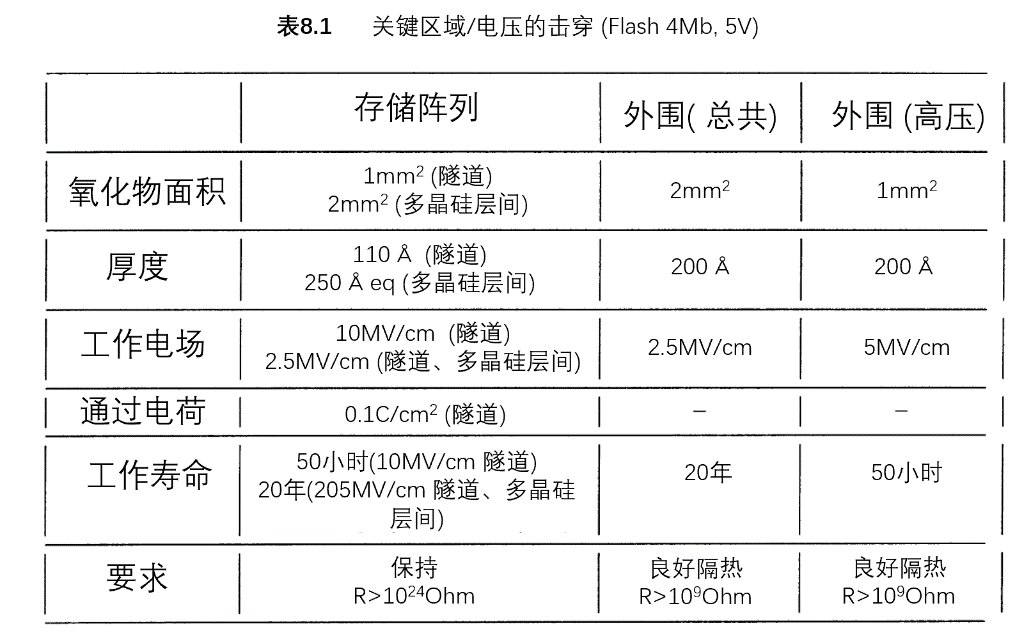
图8.2展示了一个Flash功能模型。本模型展示了复杂的Flash内部结构，然而从用户的角度上看，Flash使用较为简单（类似ROM接口的读指令和类似SRAM接口的擦除和编程指令），但实际上其内部的擦除和编程算法极其复杂，同时内部的复杂结构也让测试变得同样复杂。



## 8.2.2 Flash内的氧化物电应力（Oxide Stress）

氧化物是决定VLSI产品功能与可靠性的最关键要素之一，而对于非易失性存储器（Non Volatile Memory, NVM），它无疑是最重要的那个。

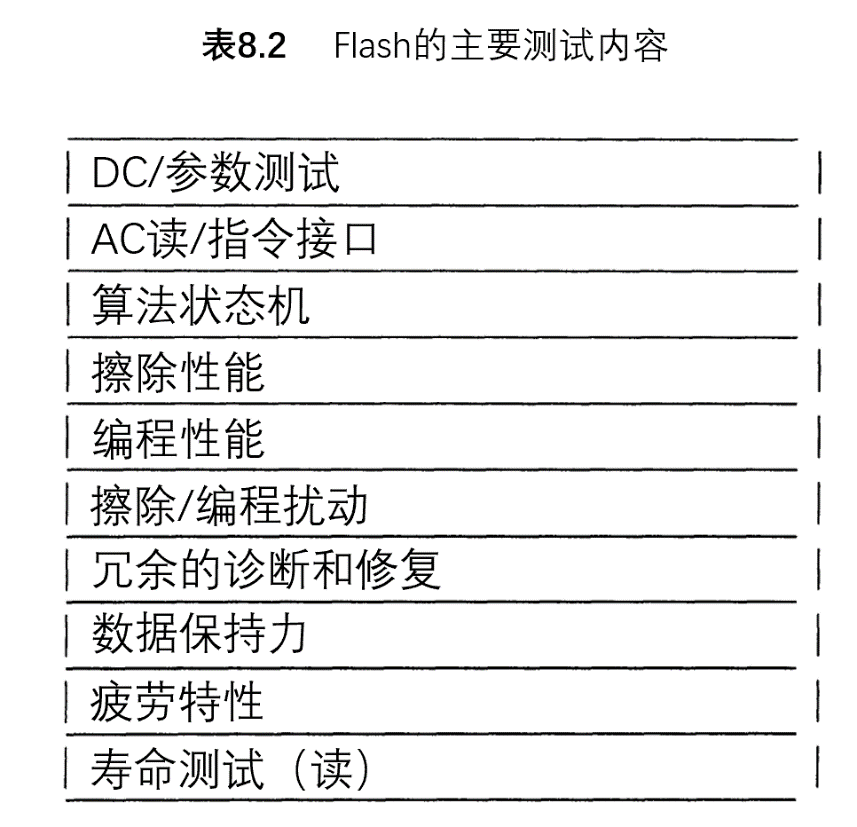
表8.1介绍了这一概念：存储阵列（array）中被薄氧化物覆盖的总面积与外围电路相当，但阵列中的隧道氧化物受到的电压（10MV/cm）要远高于工作在高电压下的外围电路（5MV/cm)。并且，外围电路的氧化物只需要保证良好阻抗（几十MΩ）来让MOS晶体管稳定切换，而阵列中的单元必须保证数据稳定存储多年。在电阻率上，这两个区域的氧化物之间有15个数量级的差别。这也解释了为什么绝大多数的可测性工具和筛选的重点都放在了存储阵列上。



## 8.2.3 Flash测试各方面

测试Flash产品必须考虑表8.2中列出的一系列内容。

Flash测试的一个特殊点在于它的擦除和编程机制比读操作慢得多：擦除或编程一个区域所需的时间在秒级，而读出相同区域的数据只需要几毫秒。相比于SRAM和DRAM这是一个很大的差别，因为它们读和写需要的时间相同。这表示对于Flash，构建一个按线性顺序独立地检查分析不同方面的测试流程是不切合实际的。相反，设计测试流程必须尽可能减少擦除/编程的周期数，并且不同测试方面的覆盖范围必须按测试流程分布。同样，缺陷的诊断和修复也可能出现在测试流程的不同步骤中。



## 8.2.4 概念测试流程

为了使读者理解接下来关于可测性工具和错误修复的内容，这里介绍一个概念性的Flash生产测试流程（图8.3）。该部分在第8.4节中有更详细介绍。

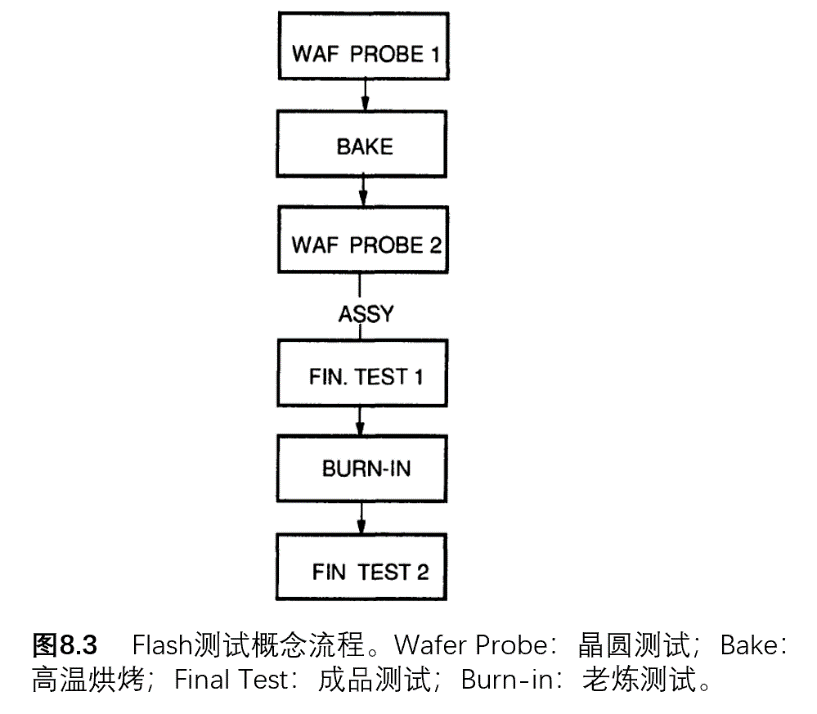
首先开展的晶圆测试涵盖了绝大多数方面：所有的DC和参数测试，读测试，指令接口的功能测试和算法状态机的功能测试，擦除/编程性能和相关的扰动测试。晶圆测试过程，通常将AC条件设置为比成品测试更宽松。

几乎整个晶圆测试过程都进行测试过程中的缺陷检测，并判断它们是否能被修复，并执行相应的修复策略。绝大部分Flash生产商都利用基于Flash的非易失性阵列来永久存储用于修复的备用行和列的地址。在这种情况下，修复在首次晶圆测试中也能进行。一些关键参数能被收集和监测来为这一过程提供短环反馈：激活和空闲状态（active, stand-by）下的Icc电流，单元电流，擦除和编程次数，编程后的阈值电压（Vt）等。原始良率（未修复），修复后的良率，硬失效的次数以及不同测试过程中修复数量都会被记录，对这些数据的分析将作为之后提升良率的主要依据。

通常会对所有可编程非易失性存储器进行250°C的高温烘烤（Bake，通常24小时或更多），以此来加速潜在的数据保持失效，然后在下一轮晶圆测试将这种类型的失效筛选出来。

在封装后，一个典型的测试流程可能包括两种温度情况（例如商用产品设置25°C和80°C），所有的方面都需考虑到最坏的边界条件。AC性能将采用最差情况模式（worst-case 测试向量）进行完整测试。

在两次成品测试（Final Test）中间通常会加入老炼测试（burn-in）来加速单个存储单元失效显现。这种失效对电压和温度敏感，并且可能在封装过程的应力下产生。通常使用125°C或稍高的温度（受限于塑料封装的特性）进行数小时测试，并且供电电压应高出标称Vcc的30-50%。



# 8.3 可测性工具

自从Flash诞生，点对点（ad-hoc）的可测性工具就被设计进产品内，其促进了技术进步和可靠性的改善。而在更新型的Flash产品中，嵌入式擦除/编程算法的引入、大量的区域组成的更大存储容量、单元内部产生擦除/编程电压等特征使得对于内部可测性工具的设计需求更加迫切。

不同测试工具可能被用于不同的目的：

* 阵列特性分析和筛选；
* 测试生产力；
* 设计可测性；
* 冗余；
* 产品差异化。

绝大多数测试工具都能被用于生产测试流程，相关的测试通常在可接受的时间范围内（远小于秒）。在产品的调试、产品/工艺的特性分析过程中，测试模式（test modes）由于时间限制更宽松，被大量使用在可靠性评估，失效分析等环节。测试电路能占到芯片面积的2-5%，其占比随着Flash尺寸的微缩和测试经验的积累而减小。

## 8.3.1 着重于存储单元和技术

在这一类别下最普遍的测试模式包括：

* 直接访问单元特性（阵列，参考单元等）；
* 生成用于快速缺陷筛选的高压电应力测试；
* 筛选耗尽或者低Vt的单元；
* 设定参考单元的Vt；
* 修改不同操作模式下的读出条件或单元偏置条件，用以调试、判断余量（margining）或者特性分析。

下文讨论了部分常用的测试工具。

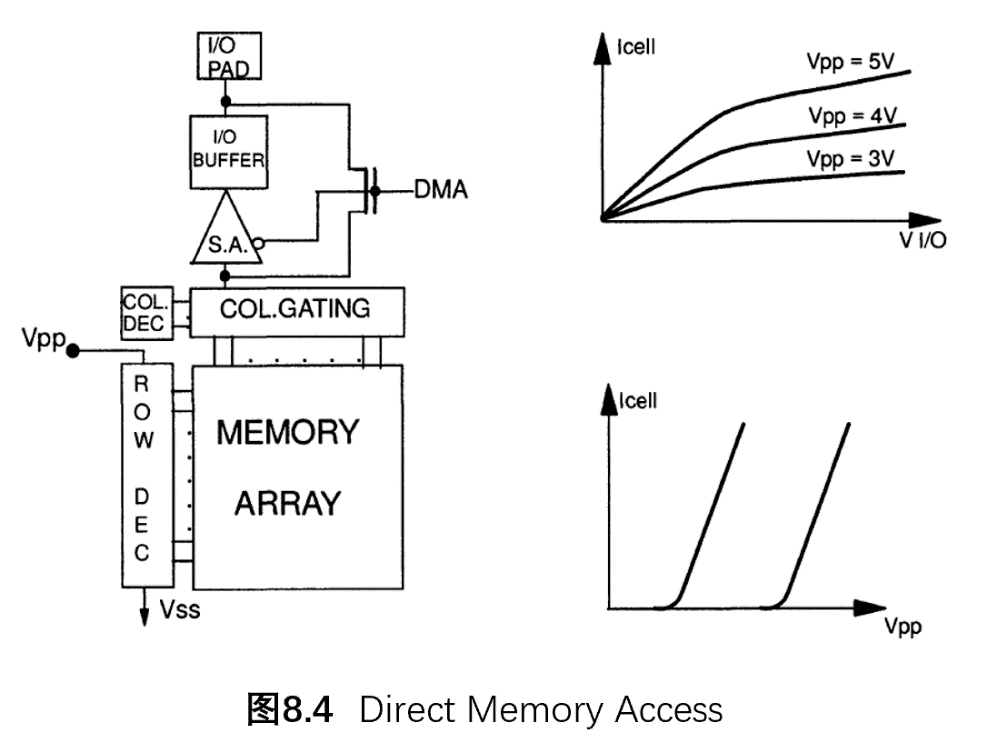
**8.3.1.1 DMA**

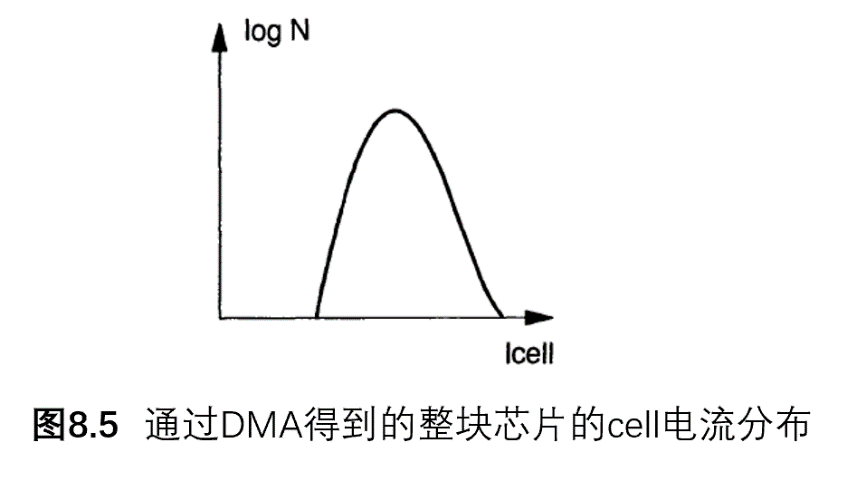
直接存储访问（DMA, Direct Memory Access）是失效分析的一个主要手段。如图8.4，一个由旁路连接到读出放大器 (Sense Amplifier) 的晶体管让I/O板能够直接连接到选定的位线。Sense Amplifier被关闭，输出缓冲变为高阻态。Vpp作为供电电压被施加到行译码器，用来大范围调整选定单元的栅极电压。这一方案可以被重复应用在全部8个，16个（或更多）I/O pin上。阵列中每个单元的特性都能在I/O pin上被观测到，如图8.4。

除了用于失效分析，DMA还能用于监测典型单元的电流，以此来调整生产或分析工艺/产品特性。阵列中单元电流的收敛分布（在UV擦除、电擦除或编程之后）是Flash器件的一个主要问题，它普遍被用于新工艺、工艺调整或者缩小的评估。

DMA的局限在于测试机本身的慢速参数测量单元：通常要花费数十毫秒。为了更好地利用DMA，一些测试机提供了快速并行参数测量单元（每个I/O有一个单元），能在几毫秒内测量16个单元的电流。然而，要得到一个Flash的完整单元电流分布（如图8.5）可能需要花费数小时。

通过寻找分布的尾部（tail），快速且实用的分析能在几分钟内完成。





**8.3.1.2 Vt测量**

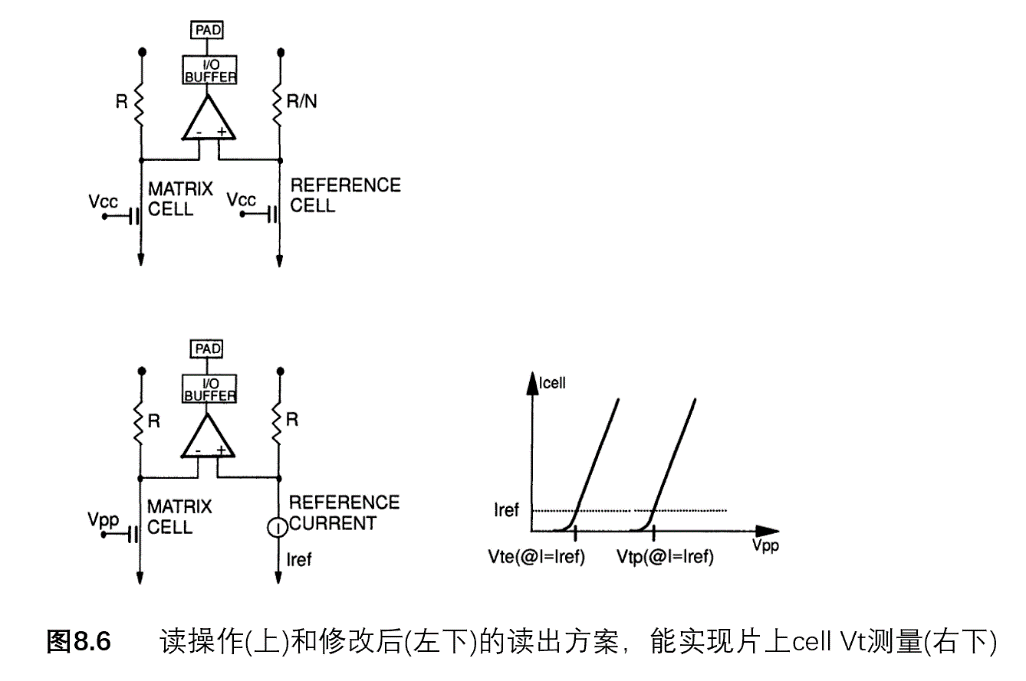
大多数NVM都采用两端sense方案（见第五章）：地址被选中的单元的电流被加在负载（load）晶体管上，其电压输出加在一个电压比较器的输入端；比较器的另一个输入来自相似结构，参考单元的电流被加在一个与矩阵端的负载成给定比例（如2）的负载上，产生电压输入进比较器。

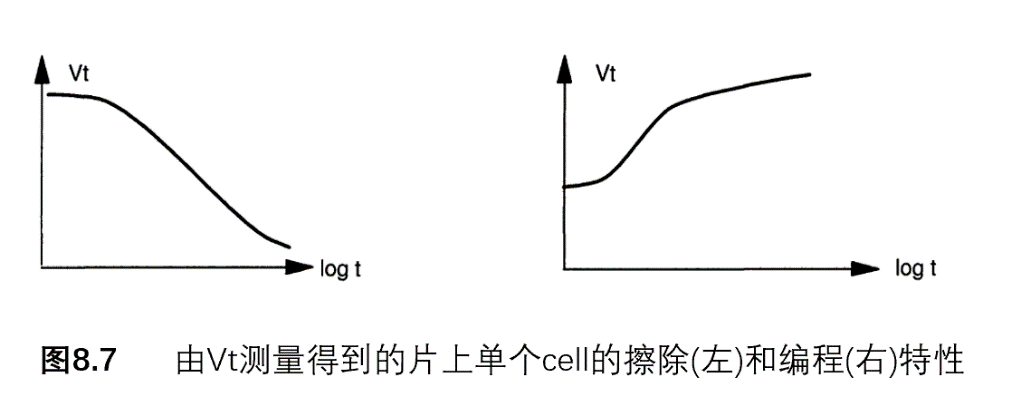
稍加修改，这个电流sense方案就能被用于测量矩阵单元的Vt。如图8.6给出的例子展示，利用一个固定的参考电流和1比1的负载。通过在栅极施加变化的电压（例如给行译码器加供电电压Vpp），给定电流条件下的Vt就能被测量（即I/O pad切换时的Vpp）。这一方法使Vt的测量和读操作能够同时进行。

在实际应用中，一个限制测量的因素是测试机的错误计数时间（error-counting time）：对于一个大存储器，Vt go-no-go测试（或者统计Vt低于指定值的单元，并生成相应的bit-map）能在几分钟内完成。如果只分析位于分布头和尾的Vt，该过程只需不到一秒。测试机提供的I/O快速失效计数功能也能提升速度。

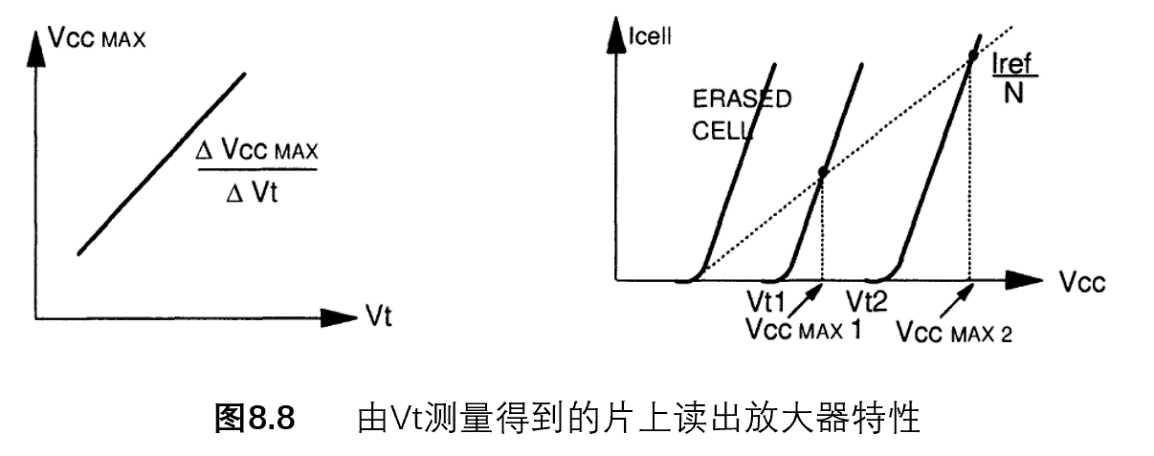
在电路层面，限制测量的因素源自译码电路的最小工作电压（约1V）。低于最小工作电压的Vt无法被测量，只能由线性外加得到。改变Iref参考电流的值能尽量避免这一限制。

Vt测量被广泛应用在生产测试中，用来确保编程、擦除、高温烘烤等后的数据裕度。用Vt测试来进行工艺/产品特性分析的一个例子如图8.7。产品中单个单元的擦除和编程特性能被计算得到，并与工艺中基本结构的单元特性进行对比。





另一个使用Vt测量的例子如图8.8。右边展示了单元的三种特性：良好擦除，良好写入，以及在中间的临界（marginal）单元。假设使用一个负载比率为N的两端sense，则虚线代表参考单元的电流除以N。通过在临界附近擦除或写入一个单元，可以得到一个Vccmax（即单元能被正确sense的最大Vcc）和其Vt的关系曲线（左图），以此来检查sensing是否如预期。



**8.3.1.3 电应力测试 (Stress Modes)**

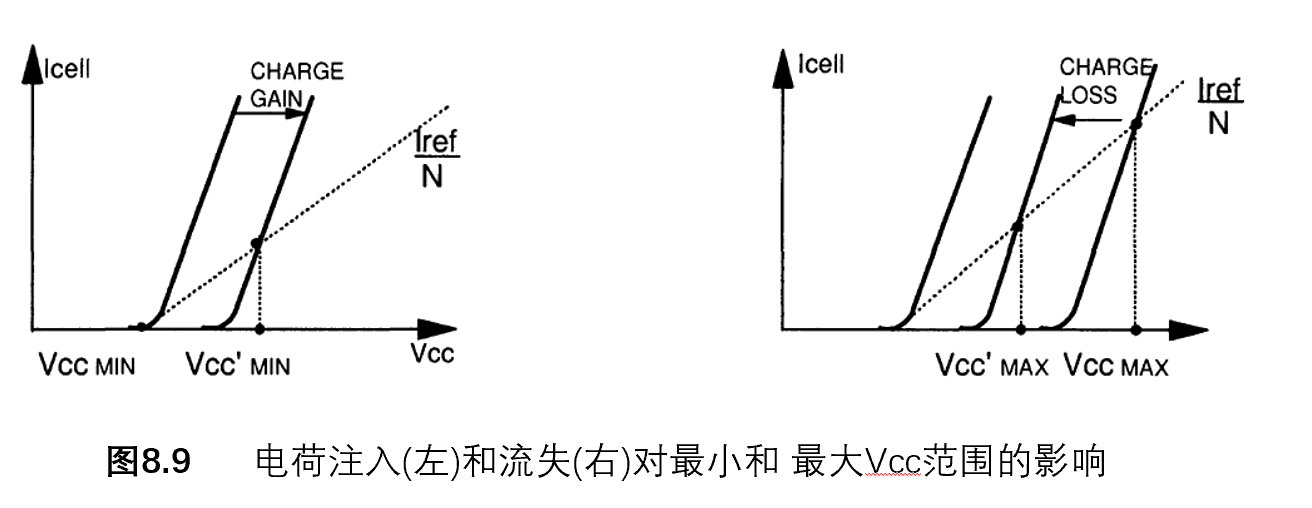
Flash的潜在失效模式包括由隧道氧化物或多晶硅层间氧化物异常造成的电荷注入或流失的失效，通常影响单个单元。这种类型的失效会因为阵列扰动（array disturb）而加重，通常与编程时加在字线和位线上的高电压stress相关。典型的失效包括：

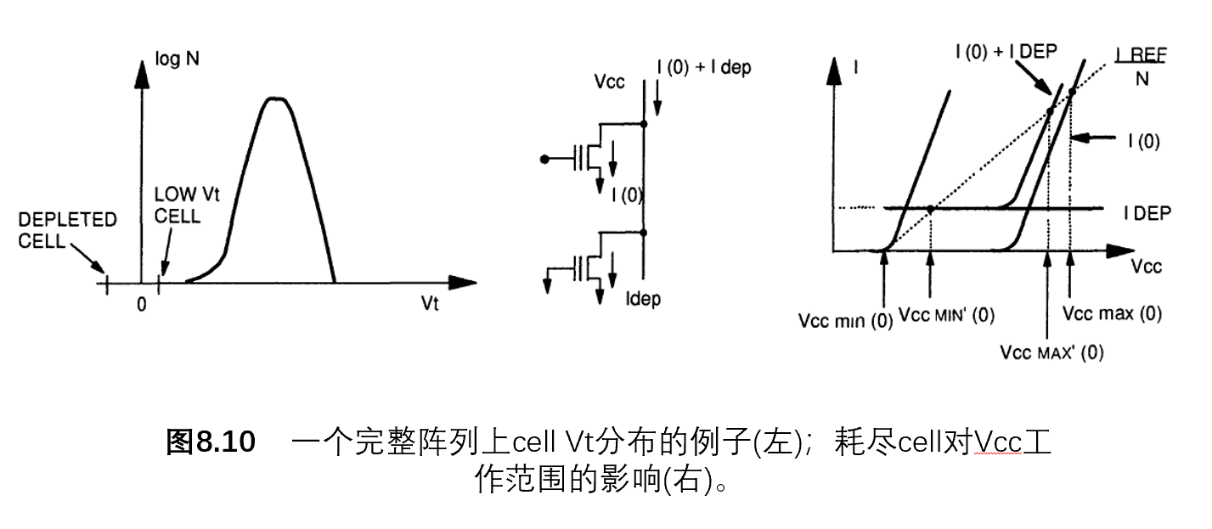
* 栅极stress造成的已擦除单元电荷注入；
* 栅极stress造成的已编程单元电荷流失；
* 漏极stress造成的已编程单元电荷流失。

Flash芯片通常包含一些测试模式用来快速筛选出扰动敏感的单元，比如：

* 栅极stress：所有字线同时加高电压（Vpp），位线开路；
* 漏极stress：所有位线加5-6V电压，字线接地。

通过检测电压对单元的Vt或者对Vccmin、Vccmax的影响，扰动敏感的单元能被筛选出来。如图8.9，一个已擦除的单元被电荷注入后，使单元能被正确读出的最小Vcc增加（左），而一个已编程的单元流失电荷后，使单元能被正确读出的最大Vcc减小（右）。





**8.3.1.4 耗尽/Low-Vt 测试**

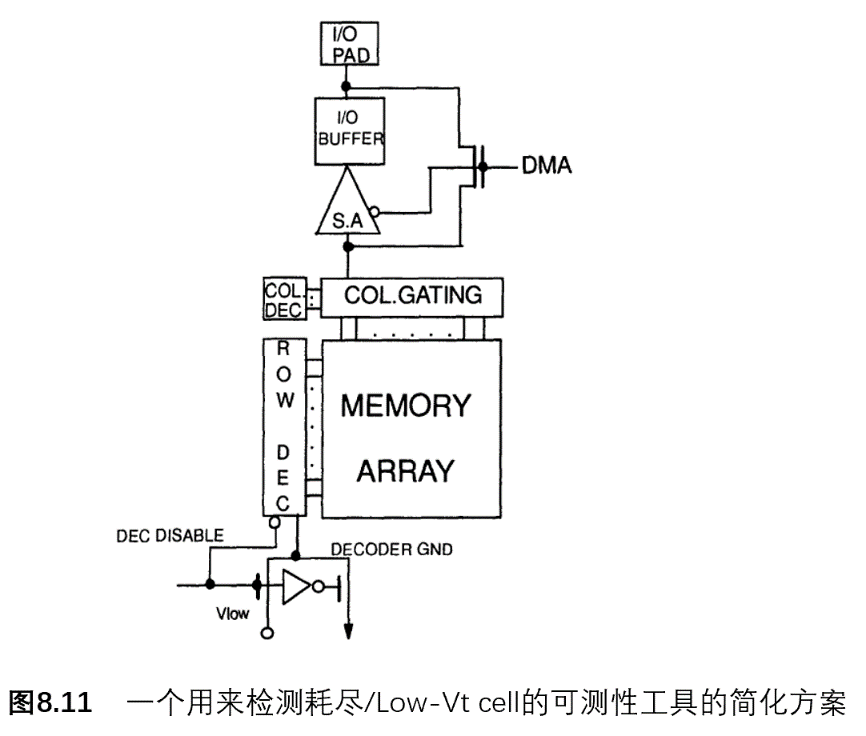
擦除后变为耗尽（depleted）的单元是NOR Flash的一个主要潜在失效原因。耗尽单元的电流能干扰同位线上（或者在按行划分的情况下，子位线上）已编程单元的读操作。图8.10（右）显示了耗尽单元对sensing的影响 ：使已编程单元能被正确sense到的最小工作电压从Vccmin(0)增加到了Vccmin‘(0)，而Vccmax(0)也减小到了Vccmax‘(0)。

Low-Vt的单元（Vt只比0高数百毫伏）也会增加硬失效、耗尽失效或增益衰减的概率。

耗尽或者Low-Vt的单元能被用于特性分析、筛选或者激活恢复算法的特定测试模式所检测到。图8.11展示了一个简化的测试方案。

耗尽测试

禁用行译码器，并“接地”到0V，则所有字线都被接地。如果DMA模块处于启用状态，就能在I/O端口测到耗尽单元的电流。如果sense处于正常工作模式，它就应该能在Vcc降到最小值，电路停止工作时检测到0；如果sense在比Vcc最小值高的电压开始检测到1，则该位线上可能有耗尽单元。同样，耗尽单元能用Vt测量模式检测到。



Low-Vt测试

禁用行译码器，并把行译码器“接地”到一个正电压（Vlow），则所有字线电压都是Vlow。然后，DMA能被用来检测选定的字线上Vt比Vlow小的单元。擦除态分布中Vt较小的部分也能用这一方式找到。

耗尽测试也能在不使用特定耗尽测试模式的情况下进行：选出阵列中的一行编程，其他行保持擦除状态。在启用DMA, 低Vcc（低于4到5V）的情况下分别选中编程行的每一个单元，如果位线上有电流，则可能是由耗尽单元造成（若改变I/O电压，则耗尽单元的电流可能被掩盖）。

同样，当sense正常工作时，一个异常的Vccmin(0)也能表明耗尽单元的存在。

以上介绍的耗尽测试只能定位被影响的位线（可能有不止一个耗尽单元的电流在累积）。用DMA或Vt测试来扫描该位线就能定位具体哪个单元已耗尽。

## 8.3.2 着重于测试生产力

由于Flash擦除和编程速度较慢，测试工程师总是需要考虑测试时间。对于第一代bulk-erasable的Flash，芯片的擦除和编程时间都在一秒左右，因此一次顺利的测试可能花费十秒左右（这些时间参照由沟道热电子注入编程和Fowler-Nordheim隧穿擦除的NOR Flash，本章的下文中也相同）。而第二代产品开始划区块块，尺寸也变大，测试时间要慢得多。对于一个有32区块的16Mb Flash，擦除时间能变为32倍，编程时间变为16倍，造成一百秒左右的总测试时间。

为了节省不必要的成本，并行测试被应用在芯片内部和外部的测试上。

并行测试通常有两个基本的特征：

* 并行编程：数个byte或word能一起被编程。该技术的局限在于编程一个bit所需要的电流量级（主流的Flash都需要几百微安）。若增加bit的数量，内部的电阻压降也增加，造成漏极电压下降，最终使编程变慢，并偏离正常的操作。这个问题对单电源供电、低Vcc（3V或以下）的产品尤其致命，因为它们需要内部电荷泵来提供漏极电压。

为了更好地引入测试模式，不同字节经常被编程为相同的内容。在一些产品中，并行编程可能作为一个特殊功能提供给用户：一个页寄存器被用来储存并行编程所用的任意内容。

* 并行擦除：数个或者全部区块被一起擦除。

最初作为一种测试模式被引入，这一功能现在广泛被作为一个必要功能提供。可以对芯片发出同时擦除任意区域组合的指令，然后由擦除算法来负责并行擦除。并行擦除的局限在于被擦除的区域所需要的源极电流（毫安范围内）。同样，这对单电源供电，低Vcc的产品的内部电荷泵尺寸也有影响。

## 8.3.3 着重于设计

Flash复杂的逻辑结构要求在设计中考虑特定的可测性特征，原因在于：

* 逻辑几乎都是顺序性的；
* 逻辑机并不直接产生pad上的数字信号，而是由逻辑机控制并激活阵列的模拟输入，进而改变阵列状态。只有阵列的状态才能在I/O pad上被读到。
* 阵列中逻辑机或者bits的异常行为（可能由不正确的设计、工艺或制造过程的缺陷造成）会变得非常难以理解和筛查。

另外，在测试阵列之前单独测试逻辑机也是一种节省测试时间的方法。作为支持，Flash测试机通常需要提供逻辑测试（Vector Testing）的功能，或者提供输入向量并比较被测结构和测试机给出的向量输出，以此来测试逻辑结构。这在之前的NVM中通常没有要求。

其他有用的可测性特征可以设计为：

* 允许以组合网络的形式测试PLA（可编程逻辑阵列）；
* 能够独立测试寄存器/计数器；
* 能够从测试机而非内部状态机来控制矩阵；
* 测量或驱动大量逻辑或模拟节点；
* 使用用户模式对应的不同逻辑或模拟条件执行擦除/编程算法，以实现筛选目的。

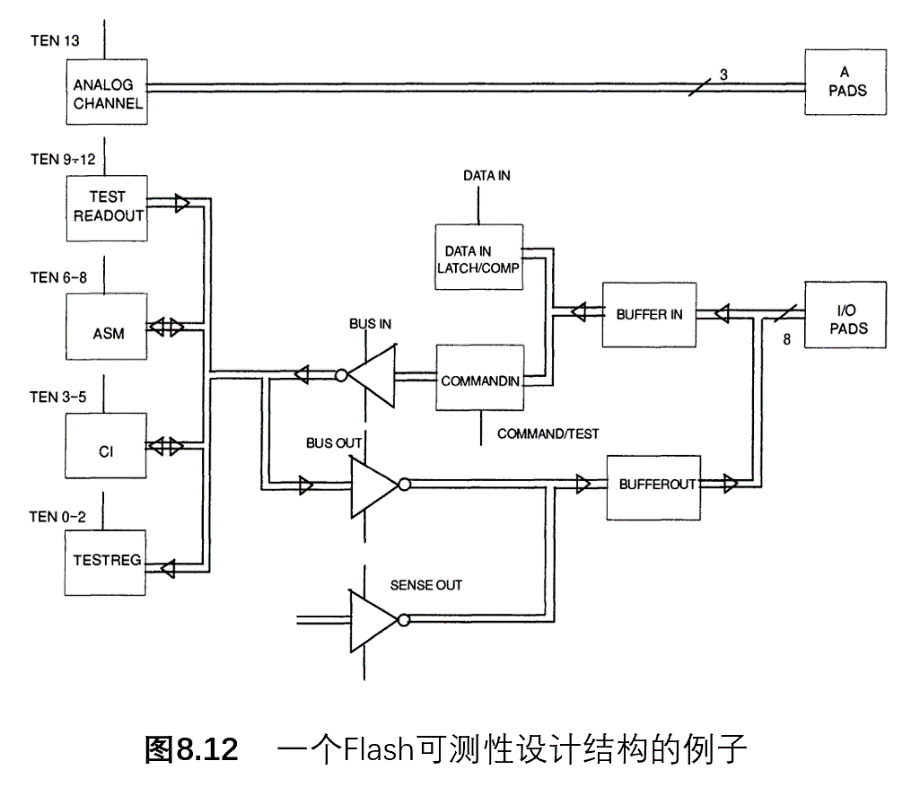
以上所有功能在调试、特性分析、生产测试和失效分析时都很有用。

## 8.3.4 Flash可测性设计：示例

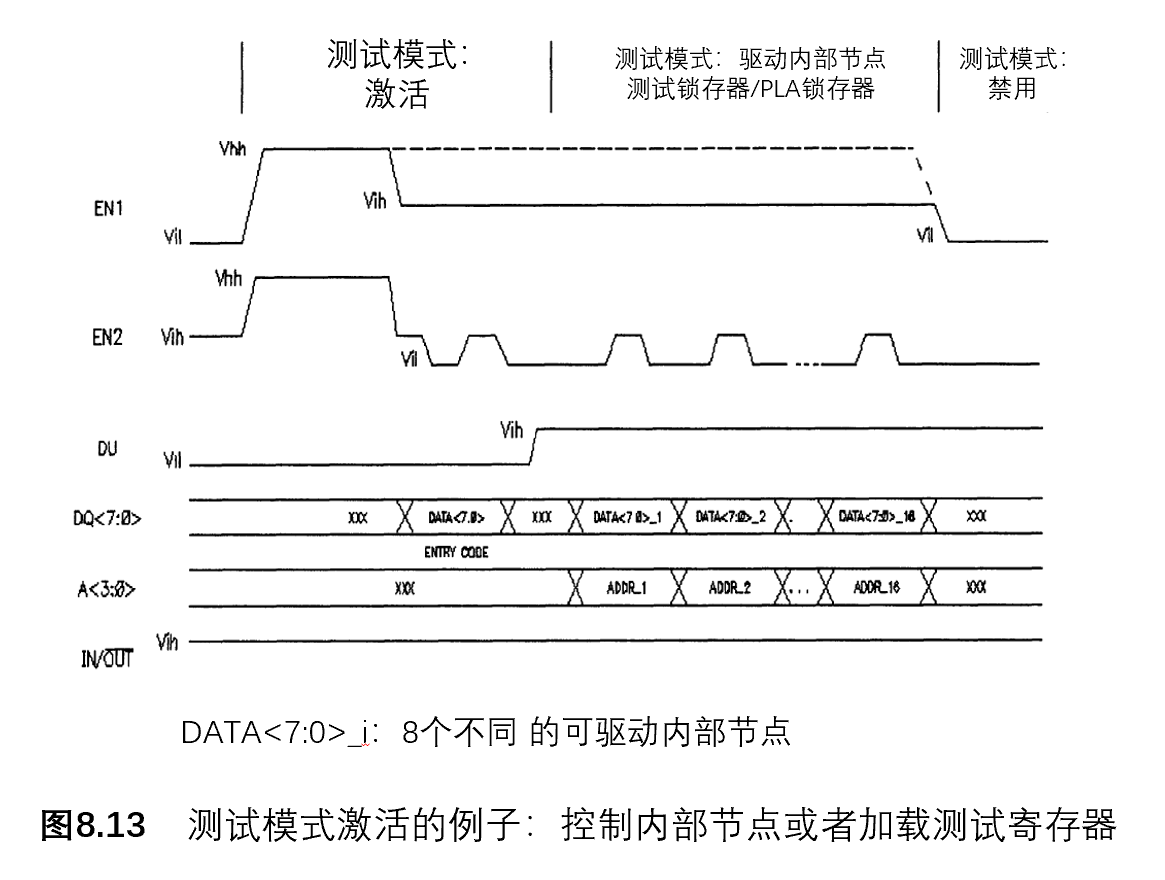
这里介绍一个Flash存储器可测性设计结构的例子。其目标是为逻辑/模拟电路和阵列提供良好的可观测性和可控性（图8.12）。

为了可测试性目标而特别增加的电路包括：

* 一根连接到I/O pads的八线双向总线；
* 连接到地址pads的三根模拟信号线路；
* 用来存储测试信号的锁存器或专门的测试寄存器；
* 用来控制连接到总线或模拟线路的多路复用器；
* 译码后用以选择多路复用器的4个地址pin；
* 一个由Vpp供电的简易D/A转换器，其被测试bits控制，用来生成多种模拟电压；
* 两个三级探测器（12V），用来为防止意外激活测试模式提供硬件保护。



在向两个输入pin提供三级之后（图8.13，8.14），测试模式的代码与用户模式的代码一样被发送到指令接口上。输出使能pin选择I/O上数据传输的方向。一个DU pin（Don‘t Use）被用来切换测试和用户模式。当测试模式开始的序列被成功接收后，指令接口允许信息被传输到测试锁存器。每个锁存器都能修改电路的行为，这样它们仍能被用于器件的其他操作上（编程，擦除，读操作等）。一个锁存器产生一个功能，这些功能可以被组合起来并产生更复杂的测试功能，因此它们能被用于实现灵活的测试模式开发。



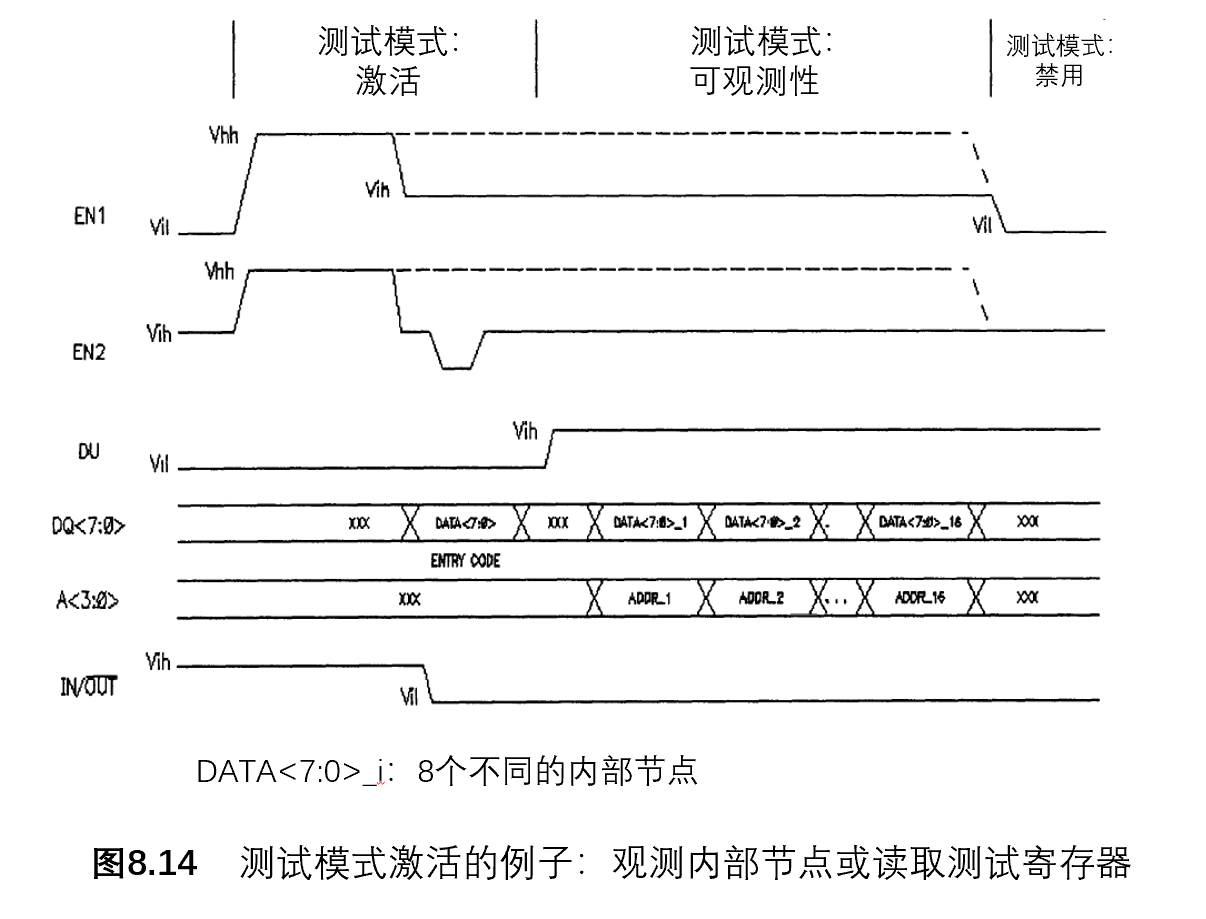
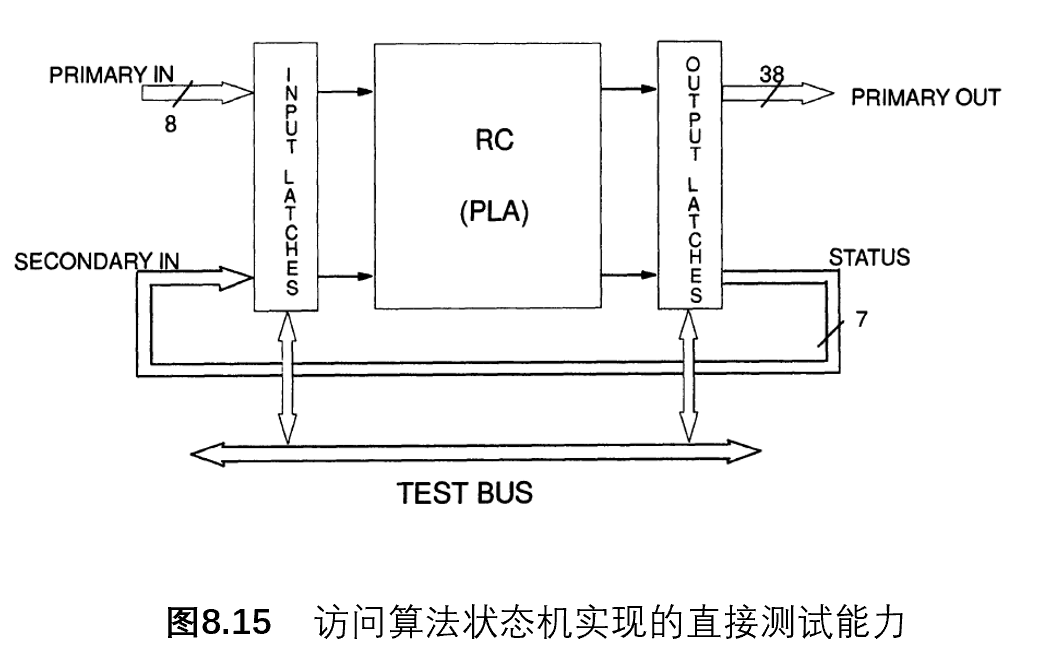


图8.13展示了驱动测试数据的例子，图8.14展示了读操作测试信息的例子。

在这个例子中，92个节点能通过不同的测试模式被观测到，128个节点被驱动并连接到测试总线（其中包括算法状态机的输入、输出），如图8.15。



# 8.4 故障修复

在所有的存储器中，阵列都拥有比外围电路大得多的晶体管密度。这是因为阵列更规则的结构和更紧凑的布局规则。另外，如8.2.2节所提到，相比于外围电路的晶体管，单元受到的stress以及相关要求更高，这造成阵列上的缺陷密度通常远高于外围电路。因此，大多数存储器利用故障修复来提升良率和可靠性。

冗余是Flash里最常用的故障修复方法。错误纠正（Error Correction）可以被用来提升良率，但它通常不用于标准的Flash。

## 8.4.1 错误纠正（Error Correction）

错误纠正更多作为在使用大容量存储器时提升板级可靠性的技术被应用，但理论上它也能被用在存储芯片内部提升良率。第五章给出了一个对错误纠正的详细分析。举个例子，一个16位Flash可以使用错误纠正码（ECC，Error Correction Code)方案来修正每个word里的单bit错误。要求是每个word加5 bits（对于阵列则是30%的开销），并在数据输出通道上加入ECC电路，用来在可能有错误的21 bits里提取16个正确的bits，以及在数据输入通道为每个要编程的word生成5个额外的bits（代价是少量die面积和访问时间延长）。Sense amplifier和列选择器同样需要增加30%。

错误纠正对良率的影响决定于缺陷产生的程度和主要的缺陷类型。例如很大程度上single bit随机错误几乎能全部修正。相反一簇bit或者字线的错误则不能被修正。

在本例中，综合起来看，ECC需要的开销为20-30%，加上数纳秒额外的访问时间，而在良率和可靠性上的收益则随缺陷的程度和类型有很大变化。

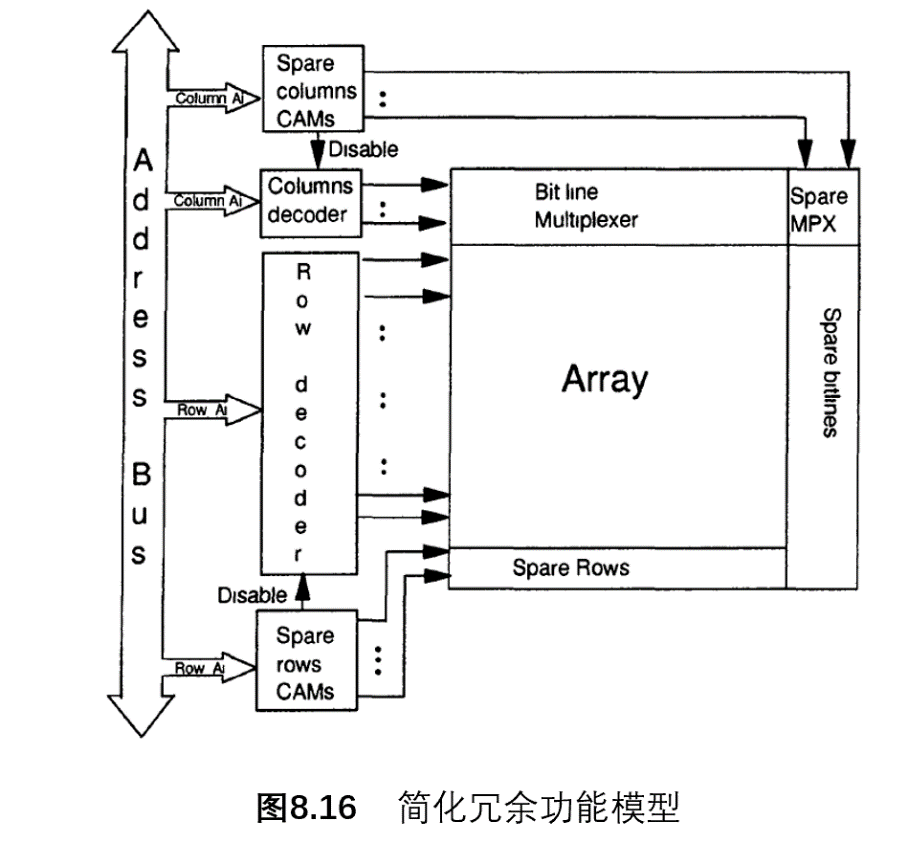
考虑到Flash优秀的可靠性和以上的分析，不难理解为什么标准的Flash通常都不使用ECC。然而，当Flash架构使用很长的字长或采用串行访问时，ECC就变得十分方便（需修正的bits对总体影响小；访问时间的牺牲不太重要）。ECC可能也会被要求应用在多级（multilevel）Flash技术上。

## 8.4.2 冗余（Redundancy）

冗余作为提升良率的方法被广泛运用在所有存储器类型中。图8.16展示了一个常被使用的方案：少量备用的字线和位线被加在矩阵阵列上，每个备用资源的访问都由CAM（Content Addressable Memory）控制。它的内容能被永久性编程。如果阵列里的一条字线被发现fail，它的地址会被编程到CAM里。然后，每次这条fail的字线被寻址的时候，CAM会产生一个“hit”信号来禁用常规译码器，并激活备用的字线。同样的方式也被应用在位线上。

为了把想要的地址永久储存在CAM里，NVM元件被采用。大多数制造商使用Flash，因为它们便于制造和测试。在Flash中不太方便的是要用激光熔断的保险丝元件（如同DRAM），这增加了制造的复杂性，或者电熔断的多晶硅保险丝，但也难以制造。这两种选项都不够灵活，因为它们只能单次编程。

在实际中，由于X8或X16的位宽，区域划分的组织结构，以及尽量避免任何性能牺牲（访问时间，电流消耗）的需要，图8.16中的方案会远远变得更复杂。有时整个区域或组的字线或位线可能都会被替换来修复大的缺陷，或者被修复的元件可能是子位线。要修复一个single bit是不实际的。



冗余在die尺寸上的消耗通常很小（百分之几）。得到的良率提升可能变化非常大：对于处在学习曲线（learning curve）早期的先进技术的大型存储器，提升可能达到十倍；而对于成熟技术的小存储器，提升则很小（百分之几）。当某个技术的learning curve有进步时，存储器普遍会微缩。这会造成致命缺陷密度的增加，并因此再一次提升冗余对良率的积极影响。

由于上述的原因，冗余被用在绝大部分Flash产品上，来减少die的花费。

冗余对可靠性没有直接的影响（自诊断-修复是一个不错的创意，但它太复杂，无法用在目前的先进技术上）。然而，拥有一些可用的备用资源让对阵列单元的分布进行严格的筛选成为可能，而这对Flash的卓越可靠性有重大贡献。

Flash产品会使用冗余这一事实对用户完全是透明的。然而对于制造商来说，冗余在测试步骤、可测性工具需求以及对测试机的要求上都有重大影响。

**8.4.2.1 诊断和修复**

冗余只能修复阵列、行译码器和列选择器里的（部分）缺陷。参数失效、数据通道失效、I/O及地址失效、算法状态机失效以及擦除/编程电路失效都无法被修复。

大部分故障检测和修复在第一次晶圆测试中完成，主要包括以下三个步骤：

1. 故障检测和故障位置的实时存储；
2. 分析出错的测试数据并决定是否以及如何修复DUT；
3. 实施冗余修复。

要支持上述步骤，测试机需要具有以下功能：

* 错误存储（error memory）；
* 一个Fail RAM用来储存失效地址列表，或者储存两个向量来标记阵列的行和列，向量中每个bit的内容都表示对应的行或列中是否有失效发生。然后这两个向量被用来进行快速诊断，并决定该部分是否能被修复。
* 一个在DUT阵列上逐位标记错误的错误捕获存储器（Error Catching Memory）；
* 在DUT阵列上标记行和列（有时是I/O）的scramble RAM；
* 专用的冗余处理器。

在每个功能步骤（读测试向量，写入测试向量，擦除区域，耗尽测试）中，一个或多个失效的bit都能被检测到，并且必须被测试机实时储存在一个error memory中，被测Flash的错误由此被标记。Error memory能累积不同测试步骤中产生的错误。在一个（或一组）测试步骤后，error memory必须被分析以诊断DUT是否能被现有备用资源修复，并决定相应的资源利用方式。

在测试流程中，备用资源仅在需要的时候被激活（当被检测到的失效若不修复就会影响到后续测试时），否则备用资源只在最后激活，此时错误寻找已经完备，修复策略也已固定：失效的地址被储存在CAM中，相关的备用行或列被永久激活。在后续的测试步骤中，DUT即完好如初地运行，用户不会知道有的地方已经被修复过。

对于故障检测需注意，要检测读操作的失效很容易，而检测擦除的失效则不然。例如某区域中有一个无法被擦除的bit。如果使用标准的“用户模式“内部擦除算法，该区域的擦除会由于”超过时间限制“而失败。在那之后，失效的bit可以被轻易读出，但区域内的其他单元则可能阈值电压变小，在接下来的编程步骤中整个区域都可能失败。该例子解释了为什么需要使用测试模式的常规擦除流程，以及为什么要监测擦除的分布和异常bits的表现，来正确并安全地检测潜在失效。

同样，编程也应该使用测试模式，来进行更严格的筛选。

行冗余同样是一个问题：在阵列中被替换的字线依然会一直留在其物理位置，而内部擦除流程的设计必须考虑避免在产品的整个生命周期中对同一区域进行多次擦除，否则可能造成单元耗尽。另外，并不是所有能被备用行修复的缺陷都能在之后的产品生命周期中一直保持良好运行：错误诊断流程必须辨识并测试有缺陷的行，来决定是修复比较安全，还是应该放弃这一部分。

**8.4.2.2 可测性工具和冗余**

为了保证测试覆盖率和冗余电路（包括NVM元件）的可靠性，并尽量用小的时间成本实现大的良率提升，设计上通常会提供一些测试工具，其中常用的工具能允许：

* 在激活前对备用资源进行功能测试，称作“shadow-made“；
* 快速编程/擦除CAM；
* 边界测试CAM（已编程或已擦除）；
* 快速检测已修复的地址；
* 禁用冗余；
* 对CAM 单元进行DMA；

最后两个相比于生产更适合工程分析。

另外，所有可用来分析矩阵的测试模式都能同样被用在备用资源的单元上。

# 8.5 生产测试

在介绍完Flash如何设计之后，带着对产品和技术的可测性的关注，现在能更详细地介绍生产测试了。

在制造商方面，生产测试是一系列的顺序测试，它足够用于筛选和保障“已知“的产品，即在可控流程和参数卡控下生产的产品，其工艺流程和设计已经发展完善，并经过了大量的性能和可靠性检验。

在晶圆测试（Wafer Probe）阶段，测试通常专注于：

* 筛选DC和总功能失效；
* 故障诊断，并反馈给工艺流程；
* 故障诊断和修复；
* 参数的数据收集，用于流程控制；
* 扫描检查可靠性；
* 快速剔除不可修复的故障。

图8.17给出了晶圆测试的一个大致流程：冗余的检测、诊断和修复分布在诸多步骤中；所有步骤都意在利用前几节中介绍的可测性工具为质量和可靠性确保足够的裕度，而非仅仅确保技术参数的规格。

在成品测试（Final Test）阶段，测试通常专注于：

* 筛选DC和总功能失效（封装后；不同温度情况）；
* 确保（以及分类）某温度范围内的AC性能；
* 全面检查是否符合规格；
* 扫描检查可靠性。

图示

描述已自动生成

## 8.5.1 DC测试

不同规格条件下的Icc电流（对双供电产品是Ipp）和输出DC电平都在晶圆和成品测试中被测试。尤其地，当电流预期在微安以下时可用的Power Down测试（按规格书或作为测试模式）不仅使晶圆测试时能迅速筛选出绝大部分失效，也能构成很大部分电路的静态（static）IDDQ测试。

## 8.5.2 功能测试 （Functional Testing）

Flash与DRAM有很大特性区别（对DRAM的测试已经有大量研究）。主要区别有：

* Flash擦除和编程的速度比读操作慢3到4个数量级；
* Flash只能按区块擦除，在UV擦除后无缺陷的阵列中所有单元都是1；
* Flash 单元提供给sense amplifier的是一个静态电流（数十微安），而不是一个小电压差（数十毫伏 ）。

因此，“order of 测试向量“的概念并不能直接应用，而应该将擦除和写操作次数应该减到最少。

在阵列单元的不同故障原因中，“电阻性”故障是目前最常见的，如固定型（stuck-at）故障（stuck-at），桥接（bridging）故障等。这些用简单的测试向量就能检测到。阵列单元中更细微的故障关系到氧化物缺陷，如第七章中所描述。

在外围电路中可能存在“电容性”故障，尤其在信号非轨到轨（rail-to-rail）的情况。例如，存在于sense amplifier和它的输入（位线）、输出信号（数据线）中；在时序链中；或者在工作于TTL电平时的输入端和I/O接收端中。特性分析必须基于产品架构来设计，来排除固有的电容耦合故障，生产测试也必须能够筛选出可能故障的产品。

一个非详尽的（总）功能失效包括：

* 寻址、译码、数据通道、I/O中的故障；
* 擦除/编程电路、指令寄存器或算法状态机中的故障；
* 单个或一些bits无法读操作/擦除/编程；
* 一对bit卡住（共享同一个“open”电极，或者悬空的栅极被短接到一起）；
* 位线开路，或短接（泄漏）到地、source line、字线或其他位线；
* 对子字线也相同，并包括短接到主字线；
* 字线开路，或短接（泄漏）到地、source line、其他字线、位线或子位线。

完整的列表要长得多，但大部分的故障可以用简单向量检测到，它只需要少量的擦除/编程次数就能轻松诊断故障类型，并反馈给工艺流程和进行修复。

其中最常用和有效的测试向量包括：

* 全1：在DUT仅用UV擦除时检测一些严重故障，或在电擦除后检测擦除故障；
* 在全1上编程对角0或矩形0：编程一小部分byte，但能够检测与位线、字线及外围电路相关的大部分故障；
* 棋盘法（Checkerboard）：CK-ODD，或（对位宽为8的数据）在所有奇数地址上写入00，偶数地址上的数据为FF。使用March序列（对每个位置：读操作FF，写入00，读操作00）来执行这两个测试向量都能有效地检测出阵列、译码电路、寻址电路、数据通道及编程电路中的stuck-at和bridging故障。

综合来看，利用点对点的向量和中间的读操作验证，Flash整体功能只用一个完整的擦除/编程/擦除周期就能测试完成。

当然，要确保全规格的性能和可靠性，还需要进行更多、更复杂的筛选（见后文）。

如果不额外花费时间，算法状态机的复杂功能很难被详细测量（见8.3.3节）。如果有点对点的测试模式可用，则可以将算法状态机作为一个逻辑块单独测量。

## 8.5.3 AC读/指令接口

交流特性分析必须提供足够用于测试并按速度等级分类DUT产品的各种最坏情况条件（测试向量、寻址序列、供电电压、温度、测试向量本身的擦除和编程条件）。互补的测试向量能被用来测试所有bit的0 1两种状态。读操作模式的所有特定类型都要被测试（地址访问时间，从芯片使能、输出使能以及二者组合开始的访问时间）。不同Vcc、不同温度的测试也很常见。

同时，Flash产品常有很小待机电流的深度下电模式，这会需要更长的“恢复时间”来读操作第一个数据，而不是通常从芯片使能开始算起的读操作访问时间。这种情况也需要被分析和测试。

除了读操作（这种操作下Flash与传统的ROM或者EPROM没区别），其他所有的操作都由指令接口（Command Interface）初始化并监测（见第五章）。这个过程中，指令和数据通过I/O pad被送进和送出Flash。按照产品规格，指令接口的AC参数也会在成品测试的擦除/编程过程中用最坏情况时序条件进行测试。

## 8.5.4 擦除/编程性能；疲劳特性

擦除和编程的次数在晶圆和成品测试中都有进行测试，并考虑到足够的裕度，来确保DUT在经历产品规定的擦除/编程次数（100K）后依然符合规格。由擦除/编程次数带来的退化在产品/技术的特性和资格认证中能得到明确。

然而，尤其是在晶圆测试阶段，重点是在于筛选出擦除后处于分布头、尾的单元，或者出现异常擦除和编程表现的单元。这样的筛选，与应用严格的stress modes（见8.3.1.3节）一起，确保了产品低缺陷水平下的疲劳特性。

## 8.5.5 可靠性

如图8.3，晶圆测试通常会在第一和第二轮测试之间加入一次高温bake：通过的第一轮测试以全0（或几乎全0）测试向量的编程结束。在Bake之后，同样的测试向量会被测试。为了更有效地筛选，编程得最差的单元的Vt裕度会在bake前被记录下来，bake之后的筛选则基于最大允许的Vt变化，而不是采用一个go-no-go读操作测试。

在成品测试中，主要会使用burn-in测试来筛选氧化物缺陷，因为它们只能在温度和电压stress的组合下才能被激活。

# 8.6 测试生产力

第8.3.2节提及了一些对Flash测试时间的考虑：假设对一个区域的擦除/编程时间为一秒，且不考虑内部可能的并行性，测试一个单元所需要的时间则是数十秒。出于这个原因，Flash制造商都使用并行测试系统来进行晶圆阶段和封装后的生产测试。并行Flash测试系统的生产力很大程度上决定于测试机的结构，以及后文中会讨论的其他方面。

## 8.6.1 测试机结构的影响

简单地从硬件来说，一台测试机由下列基本单元组成：

* 系统监测器/控制器；
* 电源；
* 参数驱动/测量单元（PMU）；
* 测试向量生成器；
* 引脚电路（Pin Electronics, PE）；
* 测试向量存储器（+向量存储器）；
* Error存储器；
* 冗余处理器。

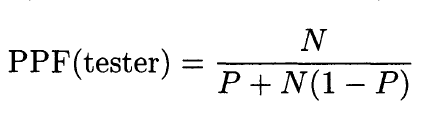
一台并行测试机能按照图8.18的方案进行制造，其中所有资源都由并行测试的N个器件（DUT）共享，除了需要在每个DUT的引脚上驱动或感应信号而被局部复制的引脚电路。测试机在从主驱动器进行驱动和在点到点进行驱动时必须有缓冲，以此来避免某个DUT由于短路而在其他器件上引起失效的情况。

另一种完全并行的测试配置为所有的N个DUT复制同样的资源块（图8.19）。第一种配置通常更便宜，而第二种则有更高的生产力。

在这两种极端配置之间存在多种中间情况。对于Flash测试，尤其在晶圆级测试，把（几乎）所有资源复制给每个DUT（tester-per-site）的测试机是目前趋势。

使用并行系统相对于固定流程的单一测试系统得到的生产力提升很大程度上决定于测试机的配置，但也决定于产品的很多其他方面。

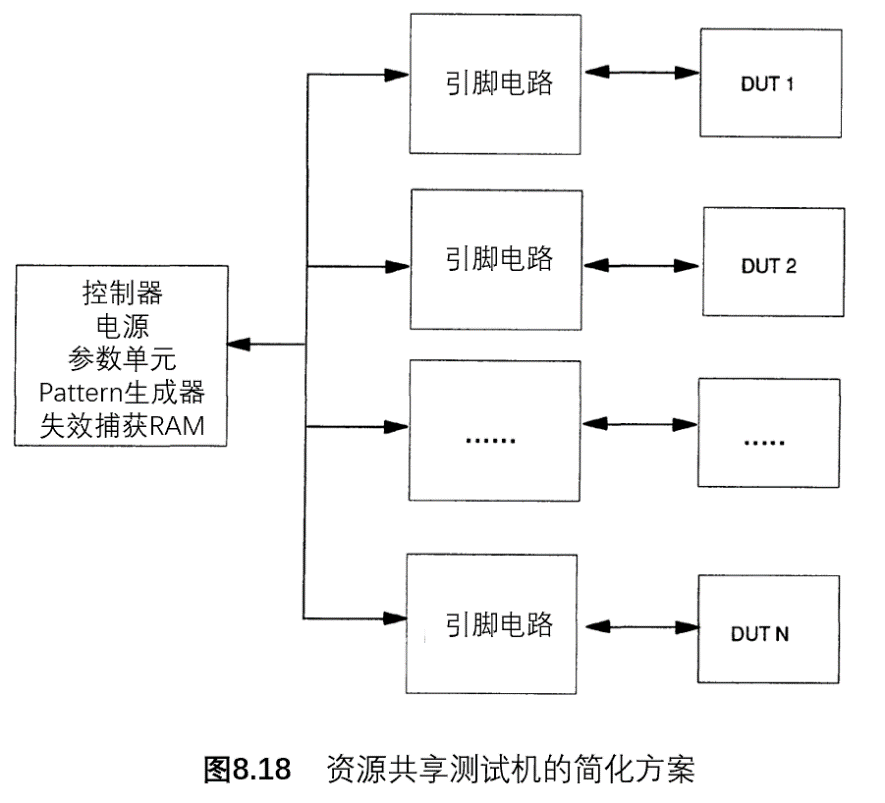
为了简化，这里只考虑测试机的影响，并假设产品方面没有生产力的影响（100%良率，性能无分散），从而能够估算对固定的测试程序，测试机理论上能得到的并行生产力因子（Parallel Productivity Factor, PPF）：

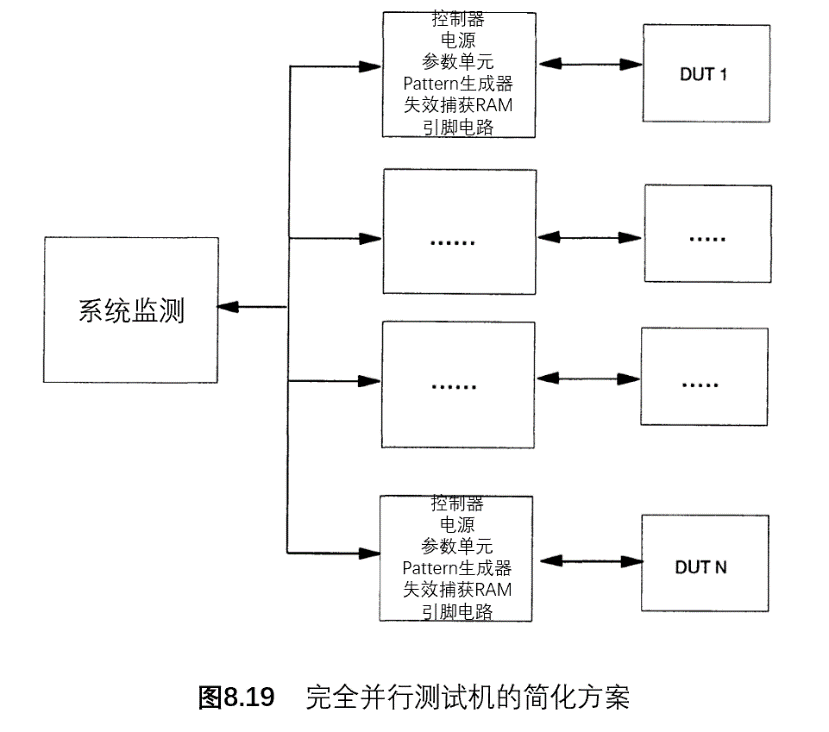


其中P是测试程序里能够并行地对N个DUT同时处理的部分，而（1-P）表示其他需要在不同DUT之间依次顺序处理的部分。

举个例子，如果PMU per site可用，参考单元的DC测试或设置就能并行运行，相应的测试时间就是P。如果测试机只有一个PMU，则上述的测试都需要对每个DUT重复、顺序地进行。

专门为Flash设计的测试机（如图8.19）能得到一个非常接近N的PPF，即理论上的极限。换句话说，测试机能同时测试N个DUT。然而，与产品相关的影响因素会很大程度上减小 PPF的值。





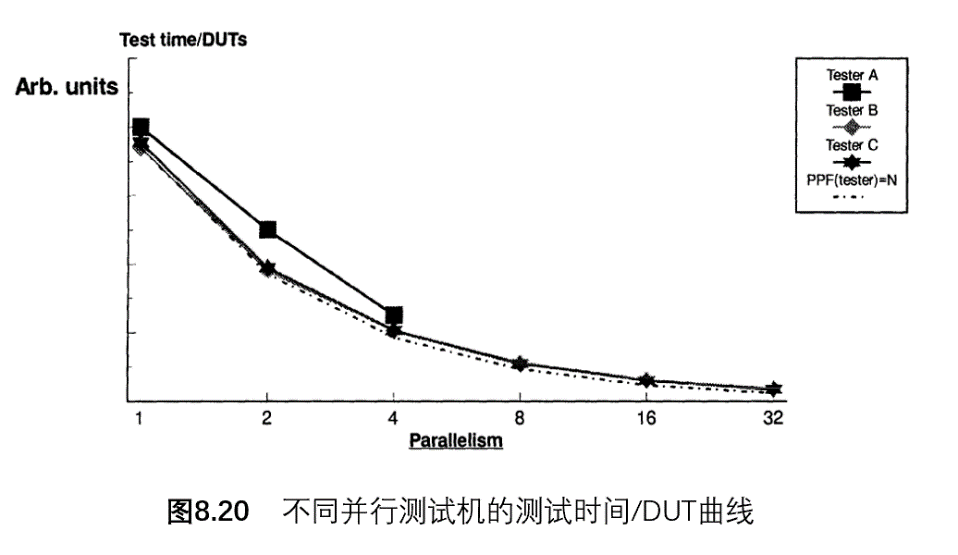
## 8.6.2 并行成品测试

为了测试封装后的产品，一个或多个机械手（handler）会连接到并行测试机。目前最先进的设备能达到非常高的并行量（例如32或64个）。

会影响并减少生产力的因素较少，它们包括：

* 良率和失效模式；
* 不同DUT之间擦除/编程时间的差别；
* 搬运时间（或者批量装载/移除N个DUT的时间）。

它们的影响无法简单用模型概括，但结果与理论值相差不远。图8.20记录了三个不同测试系统中每个DUT的平均测试时间。可以看到专门为高并行度设计的测试机（如B和C）离理论最大的PPF=N差距非常小。



## 8.6.3 EWS的并行测试

相比于成品测试，EWS（Electrical Wafer Sort）的并行测试更具挑战，并可能由于以下原因生产力较低：

* 探针卡（Probe Card）技术（考虑到Flash通常有40-60个pin）；
* 在新技术的早期阶段，良率可能较低；
* 晶圆边缘导致的低效率；
* 基于擦除和编程时间分布的测试，其中测试机会在每个地址的每个脉冲之后检查状态。与成品测试相比，EWS测试并行测试上更慢、更低效，因为成品测试通常使用内部（用户模式）的擦除/编程算法。

基于以上原因，在EWS中最多并行测试4或8个Flash，但同时16或32的探针技术也正在被引入。

在EWS中，由于测试模式和冗余的大量使用，测试机结构扮演着更为重要的角色。

# 8.7 产品特性分析

从制造商的角度，产品特性分析有如下目标：

* 在少量方面评估产品是否完全符合规格，并带有足够裕度；
* 通过测试模式来评估阵列的运行是否正确，并有足够裕度；
* 检查参数（不仅仅是列出给用户的参数，也包括一些测试能够访问的关键节点）的表现是否符合设计、工艺的要求以及物理预期；
* 评估产品是否能承受预期的工艺漂移；
* 为生产测试提取信息：最坏情况测试向量、最坏情况拉偏（corner）、限制（limits）和防护带（guard-bands）。

当然，如果检测出有问题，特性分析将反馈到设计和工艺工程师，以便调查和解决问题。

所有参数都至少要对不同Vcc和温度情况进行分析。对于双供电的产品，擦除和编程特性还需要考虑Vpp。访问时间性能需要用测试向量、地址序列、读操作模式和指定参数等来评估。

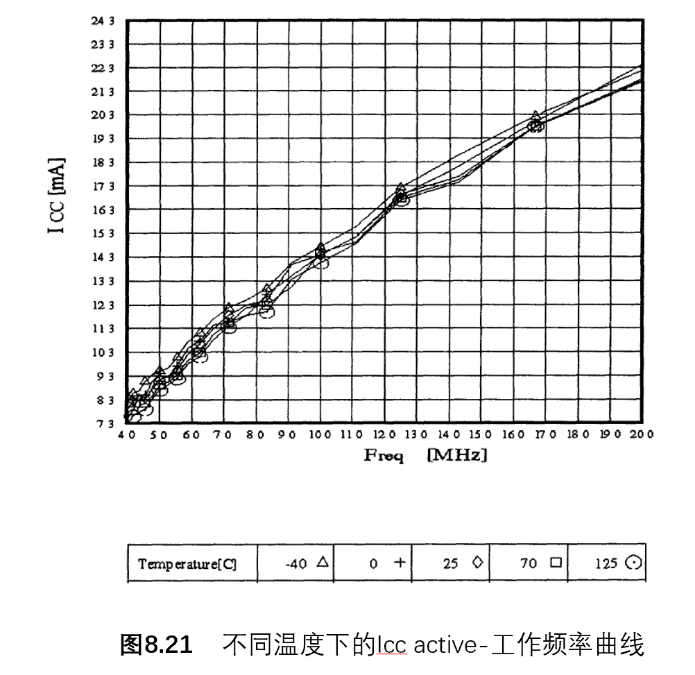
在SRAM和DRAM中，写入通道和读操作通道同样重要，这是因为写入序列被一个快速、不同步的时序链控制，导致其在不同部分之间传递时容易被改变，因此它对速度扫描和分类的要求与读操作操作相同。相反，Flash的写入指令会先被锁存，然后内部的操作会同步时钟。

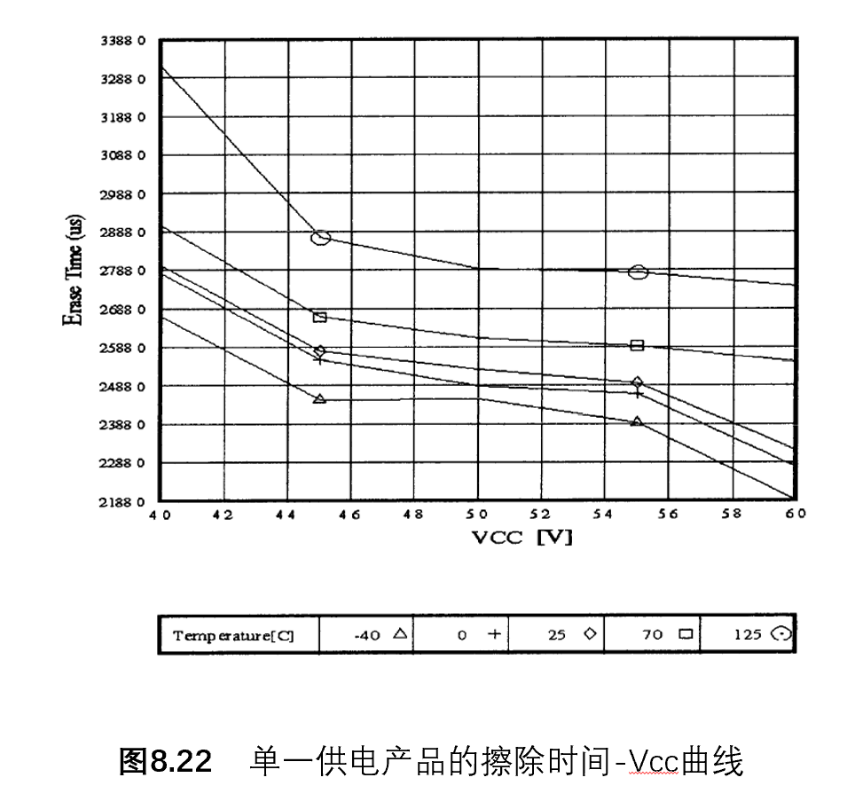
对AC写周期的分析基本上就是对指令接口的分析。一个完整的编程或擦除操作花费的时间比读操作高3-4个数量级，因此设计时会留有足够的裕度来避免这种速度差造成的问题。然而，分析读周期时要考虑的远不止速度问题。在特性分析时，需要把Flash当作ROM来分析，并考虑到阵列中的data 测试向量、架构和设计。典型的data 测试向量有棋盘型、对角型，编程的bits占不同或随机比例。但在识别I/O交互上也必须用到data 测试向量，导致sense amplifier之间或读通道信号的信息交叉或寄生耦合。

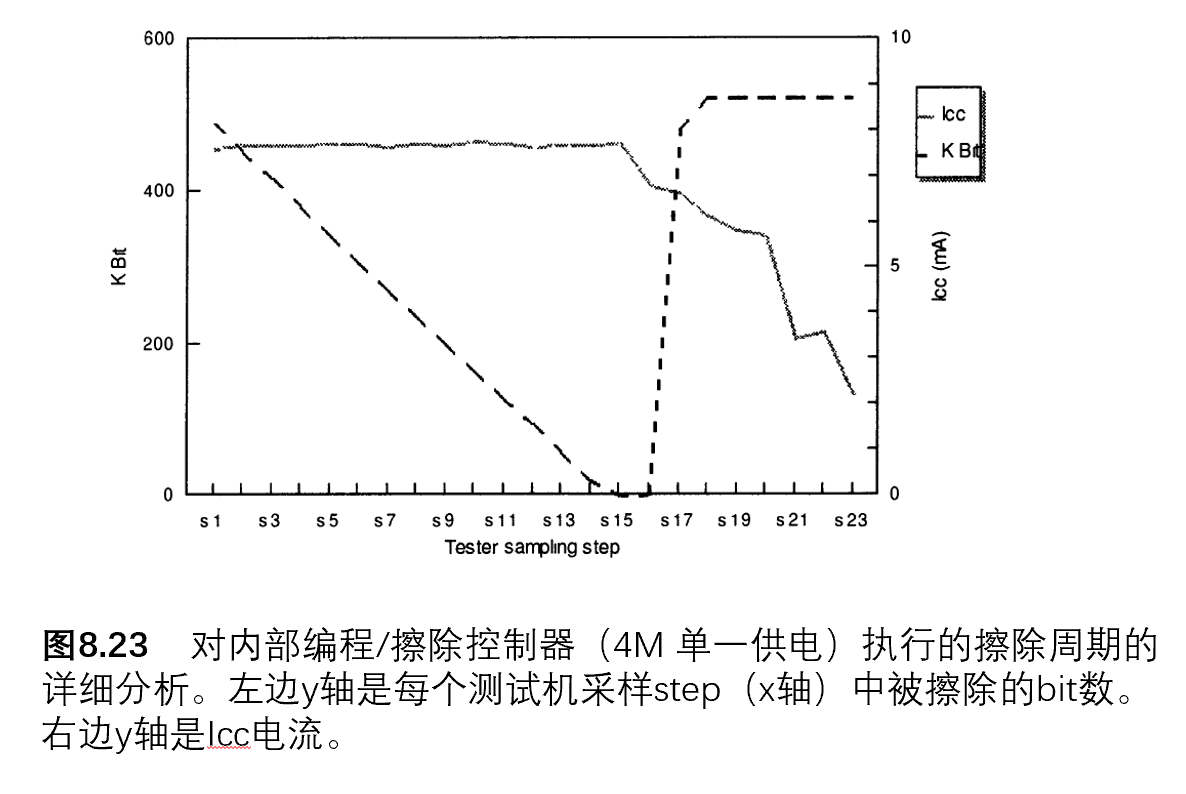
地址序列需要和data 测试向量组合使用，来凸显不同类别的问题。例如，地址补码扫描和棋盘型测试向量组合通常能作为一种输入的最坏情况。用于行访问与列访问的交互的地址序列称作蝴蝶型（从每一个单独的单元传递到相同行列的其他所有单元）。ATD (Address Transition Deteciton) 的脉冲生成能通过读操作一个附带格雷码地址序列的测试向量来检查。另外，在read-abort周期之后的读操作访问时间也需要被分析：即在一个周期内，芯片使能和地址变化的周期时间低于正常值，然后又跟随了一个正常读周期。其他的测试向量可以根据设计进行选择。

为了模拟应用上的情况，return to 1和return to 0的地址格式可以被用来反映总线上的上拉或下拉电阻。

图8.21和8.22展示了一些参数特性分析的例子：读操作模式，5.5V Vcc时的Icc-工作频率曲线，以及擦除时间-Vcc曲线。







一个完整的产品特性分析应包括对擦除/编程控制器处理的嵌入式算法的详细检查。图8.23展示了一个范例测试程序的输出：对起始阵列为FFFF的区域进行擦除。在该算法中，阵列被预先编程为0000，然后再被擦除到FFFF。测试程序通过测量从电源拉出的电流（右边y轴）和逐比特测量阵列内容的状态（左边y轴）来跟踪操作的进展。测试机每50ms采样一次bits的状态（x轴为采样step）。Byte编程操作有线性的时间消耗以及恒定的电流消耗。

在擦除过程中，阵列会很快（几个step）进入内容读出全为FFFF的状态，但内部算法需要更多时间（6个step）来保证擦除操作的足够裕度。

这样一个程序能够清晰展示内部状态机的时序分隔，同时展示了各个阶段的能耗（编程脉冲、擦除脉冲、过擦除脉冲等）。

对某个区域的各种特性分析可能先在一次短采样（10-20pcs）中尽量完成，然后再由一次更具统计性的长采样（50-500pcs）进行综合（使用更少package）。对预生产区域进行生产测试得到的参数数据需要与特性分析的数据对比，以此进行修正和性能评估。

为了更好地估计由工艺漂移造成的良率和特性变化，会刻意利用少数工艺拉偏的晶圆来对光刻（最重要）、有效区域、多晶硅尺寸、阈值电压等工艺的漂移进行评估。

# 8.8 结论

随着数代产品的更替和十多年的发展，Flash凭借卓越的性能、质量和可靠性，已经成为非常流行且易使用的存储技术。本章介绍的这些优势是如何通过在存储单元、工艺鲁棒性概念、设计可测性和可靠性、先进制造和工艺控制、详尽的特性分析和可靠性评估以及严格的生产测试筛选上的不懈努力来实现的。

本章从制造商的角度介绍了测试的各个方面，来让最终的用户对产品有更多的了解，并向读者传递这样一个信息：**一个值得信赖的制造商会用比任何用户的验收测试都严格和彻底的标准去测试每一个产品（在广义上，从设计到成品测试）。**为了用户的最大利益，我建议每个用户都将自己的测试数据与制造商的数据进行全面对比，以此实现共同利益。